(Previously Presented) Claim 1:

1 1.   In a computer system having identifiable memory address spaces which may be

2 allocated to client processes, a queue bank repository system for facilitating control of such

3 address spaces to such client processes comprising:

4   a repository of said memory address spaces wherein each memory address

5 space within said repository may be called an entry and each said entry contains a

6 value which can be either an end-of-repository space indicator or a current value

7 which indicates a next available entry, or said entry may contain a queue bank

8 descriptor, and wherein said repository has a header memory value which is either a

9 one of said current values or said end-of-repository space indicator,

10   a manager of said repository comprising

11   a functionality for generating a token wherein said token contains

12 an indication of said header memory value and for passing said token to a

13 client process if said client process passes or attempts to pass a queue bank

14 descriptor to said manager for storage into one of said memory address

15 spaces of said repository,

16   a functionality for storing said queue bank descriptor from said

17 client process into a one of said memory address spaces indicated by said

18 header memory value,

19   and wherein said manager further comprises a functionality for

20 retrieving any requested queue bank descriptor from said queue bank

21 repository upon receipt of tendered a token from any client process which

22 has previously received said token, directly or indirectly, from said

23 functionality for generating a token, and for sending said queue bank

24 descriptor to said tendering client process to enable said tendering client

25 process to retrieve a queue bank descriptor corresponding to said randomly

26 requested queue bank descriptor, said functionality for retrieving any

27   requested queue bank descriptor further comprising a functionality for

28   placing the header for the next available entry into the returned entry from

29   which said queue bank descriptor had been retrieved responsive to said

30   tendered token and which functionality is also for putting the address of the

31   returned entry into the next available entry.

(Previously Presented) Claim 2:

1   2.   The queue bank repository system of claim 1 wherein said functionality for

2   generating a token is an instruction and wherein said functionality for storing said queue

3   bank descriptor from said client process into a one of said memory address spaces

4   indicated by said header memory value is also an instruction.

(Previously Presented) Claim 3:

1   3.   The queue bank repository system of claim 1 wherein said manager further

2   comprises a functionality for reading a value in said one of said memory address spaces

3   indicated by said header value and storing said value into said header prior to or

4   contemporaneously with storing said queue bank descriptor from said client process into

5   said one of said memory address spaces indicated by said current value so that following

6   said storing, said current value shall contain a value that was in said one of said memory

7   address spaces.

(Original) Claim 4:

1   4.   The queue bank repository system of claim 1 wherein said functionality for reading

2   further comprises a functionality for clearing said value from said one of said memory

3   address spaces while storing said value into said header current value.

(Original) Claim 5:

1   5.   The queue bank repository system of claim 1 wherein said manager further

2   comprises a functionality for reading a value in said one of said memory address spaces

3   indicated by said header value and if said header value is said end-of-repository value,

4   comprising a functionality for generating a status indicating the repository is full.

(Previously Presented) Claim 6:

1   6.   The queue bank repository system of claim 1 wherein said manager further

2   comprises a functionality for reading a value in said one of said memory address spaces

3   indicated by said header value and if said header value is said end-of-repository value,

4   comprising a functionality for calling an operating system to cause said operating system

5   to allocate a new space to add to said repository and for returning a value indicative of the

6   address of the entry in said new space wherein queue bank descriptors can be stored by a

7   client process.

(Previously Presented) Claim 7:

1   7.   The queue bank repository system of claim 1 wherein said repository of said

2   memory address spaces has more than one set of said memory address spaces and uses

3   only a portion of those of said more than one sets as said memory address space within

4   said repository unless said portion becomes fully occupied and in such event, upon said

5   client process attempting to pass or passing a queue bank descriptor to said manager for

6   storage to said portion when said portion is fully occupied, a functionality extends said

7   memory address space within said repository to include another one of said one set of said

8   memory address spaces to said memory address space.

(Previously Presented) Claim 8:

1   8.   The queue bank repository system of claim 1 wherein a said token is available in a

2   plurality of different formats as determined by the specific implementation and wherein

3   said formats include, but are not limited to, pointer, offset, and entry index.

(Previously Presented) Claim 9:

1   9.   The queue bank repository system of claim 1 wherein said tokens within the

2   repository each have a size to hold any queue bank descriptor that could be generated on

3   the system.

(Previously Presented) Claim 10:

1   10.   A method of storing a queue bank descriptor from a client process into a queue

2   bank repository comprising:

3   indicating that a client process needs to store a queue bank descriptor into said

4   queue bank repository,

5   providing to said client process a token having an indication of an entry address

6   into which the queue bank descriptor is stored in said queue bank repository such that

7   the client can later retrieve the stored queue bank by returning said token to said queue

8   bank repository,

9   storing said queue bank descriptor into said entry address, and

10   removing said queue bank from the visible address space of the client process,

11   and wherein upon said return said queue bank repository a next available entry

12   address in said queue bank repository will be updated using the entry address into

13   which said queue bank descriptor had been stored.

(Previously Presented) Claim 11:

1   11.   The method of claim 10 further comprising,

2   reading from a header in said queue bank repository said next available entry

3   address location prior to providing said token to said client and wherein said storing

4   step comprises storing said queue bank descriptor into a last available entry address

5   location.

(Previously Presented) Claim 12:

1   12.   The method of claim 10 further comprising manufacturing said token to include an

2   indication of said last available entry address location into which said client queue bank

3   descriptor was stored.

(Previously Presented) Claim 13:

1   13.   The method of claim 10 further comprising manufacturing said token to include an

2   indication of said last available entry address location into which said client queue bank

3   descriptor was stored, or if the repository is full, providing an indication of fullness.

(Previously Presented) Claim 14:

1   14.   The method of claim 10 further comprising manufacturing said token to include an

2   indication of said last available entry address location into which said client queue bank

3   descriptor was stored, or if the repository is full, not providing any token until said

4   repository has an available address entry.

(Previously Presented) Claim 15:

1   15.   The method of claim 10 further comprising manufacturing said token to include an

2   indication of said last available entry address location into which said client queue bank

3   descriptor was stored, or if the repository is full, providing an interrupt to an operating

4   system.

(Original) Claim 16:

1   16.   The method of claim 15 wherein said operating system provides for more available

2   entry address locations when it receives said interrupt.

(Previously Presented) Claim 17:

1   17.   The method of claim 10 further comprising manufacturing said token to include an

2   indication of said last available entry address location into which said client queue bank

3   descriptor was stored, or if the repository is full, opening a new space of entries via a call

4   to an operating system, so that said manufacturing of said token can be accomplished with

5   an indication that said client queue bank descriptor was stored in said new space.

(Amended Hereby) Claim 18:

1   18.   A method of retrieving a queue bank by a client process from a queue bank

2   repository for storing queue bank descriptors comprising:

3   providing a token to said queue bank repository by a client process,

4   reading said token to determine an address containing a one of said queue

5   bank descriptors by said queue bank repository,

6   providing data from said address containing said queue bank descriptor to

7   said client process by said queue bank repository, and

8   establishing said retrieved queue bank in the visible address space of the

9   client process as specified by the client process.

(Amended Hereby) Claim 19:

1   19.   A method for handling invalid attempts to retrieve a queue bank by a client process

2   from a Queue Bank Repository for storing queue bank descriptors, said method

3   comprising:

4   providing a false token to said queue bank repository by said a client process,

5   reading said false token to determine an address containing said queue bank

6   descriptor by said queue bank repository

7   providing a status indicating that the token was not valid (if no deposit

8   currently exists at that token address).

(Previously Presented) Claim 20:

1   20.   A system for handling a queue bank repository system comprising at least

2   two methods, the first method, for storing a queue bank from a client process into a queue

3   bank repository comprising:

4   indicating that a client process needs to store a queue bank into said queue bank

5   repository,

6   providing to said client process a token having an indication of an entry address

7   into which the queue bank descriptor is stored in said queue bank repository such that the

8   client can later retrieve the stored queue bank, and

9   storing said queue bank descriptor into said entry address,

10   and

11   the second method, for retrieving a said queue bank that has been from a client

12   process in a queue bank repository comprising:

13   providing a token to said queue bank repository by said client,

14   reading said token to determine an address containing said queue bank descriptor

15   by said queue bank repository, and

16   providing data from said address containing said queue bank to said client process

17   by said queue bank repository.

(Original) Claim 21:

1   21.   The system of claim 19, further comprising a method for handling invalid attempts to

2   retrieve a queue bank by a client process from a Queue Bank Repository comprising:

3   providing a false token to said queue bank repository by said client,

4   reading said token to determine an address containing said queue bank by said queue

5   bank repository, and

6   providing a status indicating that the token was not valid.